AMENDMENTS TO THE CLAIMS:

This listing of claims replaces all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A method, performed by at least one device, for ~~of~~ transferring data via a communication session between a client application ~~behind a first firewall~~ and a server application ~~behind a second firewall, the method being performed by at least one device that is not behind either the first firewall or the second firewall~~, the method comprising:

assigning an identifier to the communication session;

creating at least one queue associated with the communication session;

storing data passed between the client application and the server application in the at least one queue, the data being stored using the identifier; and

receiving, from the client application, a command to obtain data in the at least one queue that is destined for the client application and that is present at a time the command from the client application is received, and receiving, from the server application, a command to obtain data in the at least one queue that is destined for the server application and that is present at a time the command from the server application is received, the command received from the client application being a hypertext transfer protocol (HTTP) command to retrieve data from the at least one device, and the command received from the server application being an HTTP command to retrieve data from the at least one device;

wherein the client application and the server application run local protocols, and the data is passed between the client application and the server application via an intermediary protocol; and

wherein the client application is behind a first firewall, the server application is behind a second firewall, and the at least one device is not behind either the first firewall or the second firewall.

2. (Previously Presented) The method of claim 1, further comprising:

creating a socket interface to at least one of the client application and the server application, data from the at least one device being transmitted through the socket interface.

3. (Previously Presented) The method of claim 1, wherein the client application and the server application are on networks that run the local protocols, and wherein conversion between the local protocols and the intermediary protocol occurs prior to passing the data through the device.

4. (Previously Presented) The method of claim 3, wherein the local protocols comprise at least one of TCP/IP and a serial protocol, the serial protocol comprising one of RS232 and RS485.

5. (Canceled)

6. (Original) The method of claim 1, wherein the identifier is associated with the at least one queue.

7. (Previously Presented) The method of claim 1, wherein the at least one device comprises a server, and the method further comprises:

performing load balancing to select the server to perform the method from among plural servers.

8. (Original) The method of claim 1, wherein the identifier is invalidated when the communication session terminates.

9. (Original) The method of claim 1, wherein the communication session comprises a telnet session.

10. (Original) The method of claim 1, wherein the communication session is effected via a Web site.

11. (Previously Presented) The method of claim 1, further comprising maintaining a session record, the session record including an identity associated with initiation of the session.

12. (Currently Amended) A system for transferring data via a communication session between a client application and a server application, the client application running on a first network ~~behind a first firewall~~ and the server application running on a second network ~~behind a second firewall~~, the system comprising:

a proxy having a socket to the client application, the proxy to convert data between a local protocol run on the first network to a non-local protocol;

an agent having a socket to the server application, the agent to convert data between a local protocol run on the second network and the non-local protocol; and

a server to enable communication between the proxy and the agent, the server containing a message queue dedicated to the communication session, the message queue for storing data transmitted during the communication session,

wherein the server is configured to receive, from the client application, data in the message queue destined for the server application, and to receive, from the server application, data in the message queue destined for the client application; ~~and~~

wherein the server is configured to receive, from the client application, a command to obtain data in the message queue that is destined for the client application and that is present at a time the command from the client application is received, and to receive, from the server application, a command to obtain data in the message queue that is destined for the server application and that is present at a time the command from the server application is received, the command received from the client application being a hypertext transfer protocol (HTTP) command to retrieve data from the server ~~at least one device~~, and the command received from the server application being an HTTP command to retrieve data from the server ~~at least one device~~; and

wherein the client application and the first network are behind a first firewall, the server application and the second network are behind a second firewall, and the server is not behind either the first firewall or the second firewall.

13. (Cancelled)

14. (Currently Amended) The system of claim 12, wherein~~, when data is present for the client application,~~ the proxy obtains the data from the message queue and passes the data to the client application.

15. (Cancelled)

16. (Currently Amended) The system of claim 12, wherein~~, when data is present for the server application,~~ the agent obtains the data from the message queue and passes the data to the server application.

17. (Currently Amended) A machine-readable medium stores instructions for use in transferring data via a communication session between a client application ~~behind a first firewall~~ and a server application ~~behind a second firewall~~, the instructions being executable

by ~~a~~ at least one machine ~~that is not behind either the first firewall or the second firewall~~, the instructions for causing the at least one machine to:

assign an identifier to the communication session;

create at least one queue associated with the communication session;

store data passed between the client application and the server application in the at least one queue, the data being stored using the identifier; and

receive, from the client application, a command to obtain data in the at least one queue that is destined for the client application and that is present at a time the command from the server application is received, and receive, from the server application, a command to obtain data in the at least one queue that is destined for the server application and that is present at a time the command from the server application is received, the command received from the client application being a hypertext transfer protocol (HTTP) command to retrieve data from the at least one machine ~~device~~, and the command received from the server application being an HTTP command to retrieve data from the at least one machine ~~device~~;

wherein the client application and the server application run local protocols, and the data is passed between the client application and the server application via an intermediary protocol; and

wherein the client application is behind a first firewall, the server application is behind a second firewall, and the at least one machine is not behind either the first firewall or the second firewall.

18. (Original) The machine-readable medium of claim 17, wherein the intermediary protocol is different from the local protocols.

19. (Original) The method of claim 1, wherein the intermediary protocol is different from the local protocols.

20. (Original) The method of claim 1, wherein the intermediary protocol is a same protocol as the local protocols.

21. (Previously Presented) The machine-readable medium of claim 17, wherein the intermediary protocol is a same protocol as the local protocols.

22. (Previously Presented) The machine-readable medium of claim 17, further comprising instructions to:

create a socket interface to at least one of the client application and the server application, data from the at least one machine being transmitted through the socket interface.

23. (Previously Presented) The machine-readable medium of claim 17, wherein the client application and the server application are on networks that run the local protocols, and wherein conversion between the local protocols and the intermediary protocol occurs prior to passing the data through the at least one machine.

24. (Previously Presented) The machine-readable medium of claim 23, wherein the local protocols comprise at least one of TCP/IP and a serial protocol, the serial protocol comprising one of RS232 and RS485.

25. (Original) The machine-readable medium of claim 17, wherein the identifier is associated with the at least one queue.

26. (Previously Presented) The machine-readable medium of claim 17, wherein the at least one machine comprises a server, and the machine-readable medium further comprises instructions to:

perform load balancing to select the server.